

2

Classificação dos dados volumétricos

Para que a visualização volumétrica possa ser realizada utilizando a renderização direta de volumes, é necessário definir um *modelo óptico* (Max, 1995) que descreva como uma unidade de volume emite, reflete e absorve um raio de luz, ou espraia o raio refletido.

Baseando-se nesse modelo, um campo escalar associado aos dados volumétricos é mapeado para valores de cor e opacidade, por meio de uma *função de transferência*. É a função de transferência a responsável por classificar e selecionar as características desejadas para serem visualizadas. Essa etapa é conhecida como *classificação* do volume. A classificação normalmente é feita atribuindo-se baixa opacidade (alta transparência) aos valores que não são importantes e alta opacidade (baixa transparência) ao que deve ser visualizado. Além disso, são atribuídas diferentes cores para cada intervalo do campo escalar. Por exemplo, supondo que se quer visualizar a variação de temperatura no interior de uma peça mecânica, representada por um modelo de elementos finitos, pode-se atribuir uma escala de cores aos valores de temperatura (p. ex., azul para as baixas temperaturas e vermelho para as altas). Se a variação das temperaturas mais altas for a característica que se deseja visualizar, pode-se atribuir opacidade alta para esses valores e baixa para os outros.

A cor final em cada ponto da imagem gerada no plano de projeção é computada integrando-se as contribuições dos valores de cor e opacidade, mapeados pela função de transferência, para cada unidade de volume ao longo do raio que passa pelo ponto do plano na direção de projeção.

Nas próximas seções são discutidos os principais modelos ópticos utilizados para renderização direta de volumes e as principais abordagens para a utilização da função de transferência na classificação de volumes representados por malhas de tetraedros lineares.

2.1. Função de transferência

A função de transferência é uma função matemática que mapeia os valores de propriedades dos dados volumétricos para propriedades ópticas, como cor e opacidade (ou o coeficiente de extinção, apresentado na próxima seção). Para um campo escalar $s : (x, y, z) \mapsto F(x, y, z)$, utilizando o sistema de cores RGB, tem-se $FT : s \mapsto (R(s), G(s), B(s), \alpha(s))$, onde α representa a opacidade (Figura 6).

Essa função é fundamental para a classificação de volumes, pois é ela que transforma os dados abstratos em características que podem ser visualizadas. A função de transferência pode ser especificada pelo usuário ou gerada por métodos semi-automáticos (Kindlmann & Durkin, 1998), e ser tão complexa quanto se queira. Por isso, a função de transferência é geralmente implementada como uma tabela de valores, que é consultada pela aplicação (*look-up table*) ou convenientemente definida por segmentos lineares (*piecewise linear segments*) (Williams & Max, 1992; Williams et al., 1998).

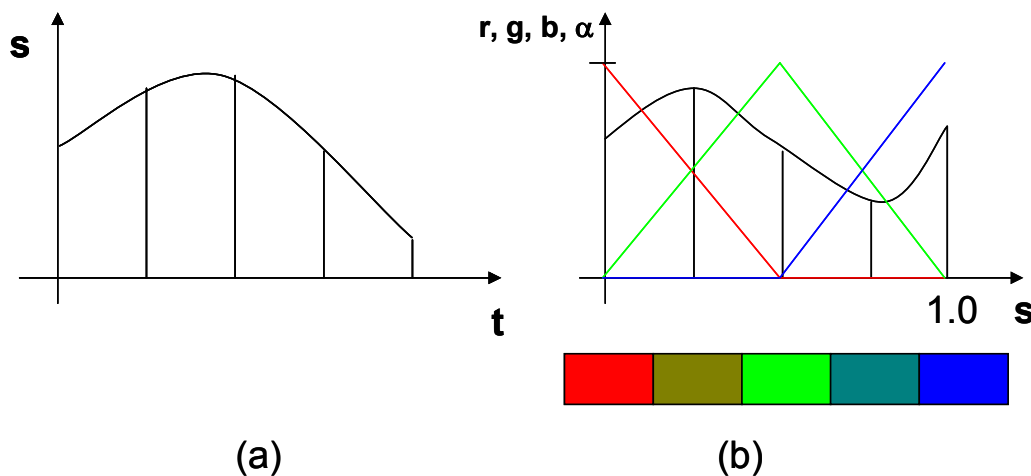


Figura 6 – (a) Valor de um campo escalar ao longo de um raio com parâmetro t . (b) Função de transferência definida para valores escalares. Os escalares são mapeados para valores de cor (r, g, b) e opacidade α .

É possível também utilizar funções de transferência multidimensionais (Kniss et al., 2002), ou seja, que são definidas em função de mais de uma propriedade relacionada aos dados volumétricos. A cor e a opacidade podem ser definidas, por exemplo, em função de um campo escalar e o gradiente deste

campo: $FT : (s, \vec{V}s) \mapsto (R(s, \vec{V}s), G(s, \vec{V}s), B(s, \vec{V}s), \alpha(s, \vec{V}s))$. A opacidade bidimensional foi recentemente utilizada com sucesso para a visualização de dados sísmicos, em (Silva, 2004), permitindo extrair características importantes desses dados que não poderiam ser visualizadas com uma função de transferência unidimensional.

2.2. Modelos ópticos

Os modelos ópticos utilizados para a geração de imagens com a renderização direta de volumes baseiam-se em modelos físicos criados a partir da simplificação da Teoria de Transporte Radiativo (Krüger, 1991). Assume-se que o meio é formado por partículas que podem emitir, absorver ou refletir raios luminosos, e a intensidade da luz resultante em uma direção é proporcional à densidade de partículas ao longo de um raio que atravessa o volume. Alguns modelos foram inicialmente criados para a visualização de nuvens, como o de (Blinn, 1982), que considera a absorção e a reflexão da luz no meio, assim como a geração de sombras. Entretanto, os principais modelos ópticos para a visualização interativa baseada na renderização direta de volumes consideram que as partículas do meio apenas absorvem e/ou emitem radiação luminosa. Max (1995) examina alguns desses modelos.

O modelo óptico de Williams & Max (1992) é provavelmente o mais utilizado, e será apresentado a seguir, seguindo a abordagem de Max (1995). A idéia desse modelo se baseia na simplificação de modelos realistas de nuvens, reduzindo-os ao mínimo necessário para permitir que a estrutura interna do volume seja visualizada.

Inicialmente, pode-se considerar uma unidade de volume diferencial, com uma densidade ρ de partículas, em que cada partícula absorve toda a intensidade luminosa que a atinge (Figura 7). Todas as partículas são esferas idênticas de área projetada A e o volume tem seção transversal de área E e espessura ds . Dessa forma, o número de partículas é $N = \rho.E.ds$. Assumindo que as projeções de duas partículas na direção de um raio perpendicular à seção transversal não se sobrepõem, então a fração da intensidade luminosa que é obstruída pelas partículas é igual a $N.A/E = \rho.A.ds$. O fator $\tau = \rho.A$ é chamado *coeficiente de*

extinção (ou densidade óptica) e está relacionado à taxa de absorção da luz incidente sobre o volume e varia de acordo com o comprimento de onda da luz. Entretanto, por simplicidade, normalmente se considera que o coeficiente de extinção é o mesmo para todos os comprimentos de onda.

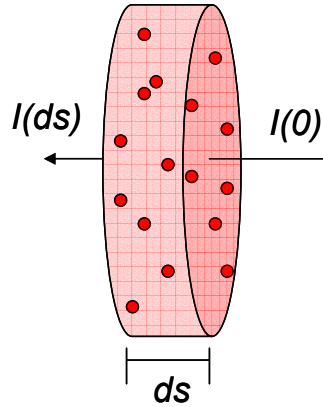


Figura 7 – Volume diferencial composto por partículas que absorvem a intensidade de um raio $I(s)$, ao longo da espessura ds .

Isso resulta na seguinte equação diferencial:

$$\frac{dI(s)}{ds} = -I(s) \cdot \tau(s), \quad (2.1)$$

cuja solução é:

$$I(s) = I(0) \cdot e^{-\int_0^s \tau(t) dt} \quad (2.2)$$

A opacidade α de um volume de espessura s é, então, definida como:

$$\alpha = 1 - e^{-\int_0^s \tau(t) dt} \quad (2.3)$$

É importante observar que a opacidade é dependente da distância percorrida pelo raio, e varia no intervalo $[0,1]$ (objeto totalmente transparente ou opaco, respectivamente), enquanto que o coeficiente de extinção é uma característica diferencial e varia no intervalo $[0,\infty)$.

Se, ao invés de absorverem a luz incidente, as partículas apenas emitirem luz, com intensidade C por unidade de sua área projetada, a fração da intensidade gerada pelo volume diferencial será igual a $C \cdot N \cdot A / E = C \cdot \rho \cdot A \cdot ds = C \cdot \tau \cdot ds$, e a equação diferencial resultante será:

$$\frac{dI(s)}{ds} = C(s) \cdot \tau(s) \quad (2.4)$$

Assim como o coeficiente de extinção, a intensidade emitida por cada partícula também varia para os diversos comprimentos de onda. No sistema de cores *RGB* (*vermelho, verde, azul*) (Foley et al., 1997), poderá haver um valor diferente para cada componente.

Finalmente, se for assumido que cada partícula tanto absorve quanto emite intensidade luminosa, as eq. (2.1) e (2.4) podem ser combinadas, resultando na seguinte equação diferencial:

$$\frac{dI(s)}{ds} = C(s) \cdot \tau(s) - I(s) \cdot \tau(s) \quad (2.5)$$

Resolvendo para uma distância D , e um raio de parâmetro t , obtém-se a solução da eq. (2.5), que é conhecida como *integral de renderização de volume*:

$$I(D) = I_0 \cdot e^{-\int_0^D \tau(t') dt'} + \int_0^D C(t) \cdot \tau(t) \cdot e^{-\int_t^D \tau(t') dt'} dt \quad (2.6)$$

Essa integral composta por uma soma de dois termos pode ser interpretada considerando-se um raio com intensidade inicial I_0 , que parte de uma posição atrás do volume visualizado na direção do observador. As posições que não estão no interior do volume não oferecem contribuição, o que implica que as intensidades nessas posições são iguais às dos pontos de entrada e saída do raio. O valor I_0 representa a intensidade de fundo. Essa intensidade é absorvida ao longo da porção do raio que atravessa o volume, o que é representado pelo primeiro termo da eq. (2.6). O segundo pode ser interpretado da seguinte maneira: cada partícula, ao longo do raio, emite intensidade luminosa, que é absorvida pelas partículas do volume que estão à sua frente. A integração das contribuições define a intensidade relativa ao volume, que é, então, adicionada à intensidade de fundo. No sistema RGB, essa integral deve ser resolvida para cada uma das componentes.

Wilhelms & Van Gelder (1991) utilizam o termo $g(t) = C(t) \cdot \tau(t)$ como uma propriedade óptica do volume. Assim, duas propriedades devem ser especificadas na função de transferência, $g(t)$ e $\tau(t)$, simplificando a eq. (2.6) para

$$I(D) = I_0 \cdot e^{-\int_0^D \tau(t') dt'} + \int_0^D g(t) \cdot e^{-\int_t^D \tau(t') dt'} dt \quad (2.7)$$

Uma desvantagem dessa abordagem é que o aumento do coeficiente de extinção faz com que a imagem fique mais escura, enquanto que no modelo de Williams & Max (1992) há um balanceamento da intensidade emitida pelas partículas pelo coeficiente de extinção.

No caso geral, não se conhece uma solução analítica para a integral de renderização de volume. Porém, em alguns casos específicos, é possível obter uma solução analítica, como nos casos em que as variações do coeficiente de extinção e da intensidade emitida são constantes ou lineares.

Se o coeficiente de extinção e a intensidade emitida forem constantes, a solução para a integral de renderização de volume é a seguinte:

$$I(D) = I_0 e^{-\tau D} + C \int_D^0 -\tau e^{-\tau t} dt = I_0 e^{-\tau D} + C(1 - e^{-\tau D}) \quad (2.8)$$

que equivale a

$$I(D) = I_0(1 - \alpha) + C\alpha \quad (2.9)$$

Dessa forma, a solução da integral corresponde à composição (Porter & Duff, 1984) da cor C , com opacidade α , sobre um fundo de cor I_0 . Se o intervalo em que o raio atravessa o volume for dividido em fatias muito pequenas, como intervalos de *somas de Riemann* (Malta et al., 2002), e a cada fatia forem atribuídas cor e opacidade constantes, a integral pode ser resolvida numericamente pela sucessiva composição, de trás para a frente (ou, alternativamente, de frente para trás), das contribuições de cada fatia. A intensidade total ao longo de um raio pode ser, então, aproximada por:

$$I = \sum_{i=0}^n \alpha_i C_i \prod_{j=0}^{i-1} (1 - \alpha_j) \quad (2.10)$$

Assim, uma forma simplificada de resolver a integral consiste em discretizar a função de transferência em pequenos pedaços com cor e opacidade constantes e realizar a composição. Uma característica da eq. (2.9) é que ela é balanceada, ou seja, a cor final nunca será maior que um valor máximo, que, convenientemente, pode ser definido como 1,0.

O termo $C \cdot \alpha$ da eq. (2.9), que resulta da integração do segundo termo da eq. (2.6), é chamado de *cor associada* (Blinn, 1994) e corresponde à cor de um segmento pré-multiplicada pela opacidade. Wittenbrink et al. (1998) demonstram que a interpolação da cor e da opacidade separadamente resulta em erros visuais

(conhecidos como “artefatos”), e Blinn (1994) mostra que a utilização de cores associadas simplifica a composição e permite a associatividade.

Para a variação linear do coeficiente de extinção e da intensidade emitida, a solução analítica (Williams & Max, 1992; Moreland & Angel, 2004) é mais complexa e de maior custo computacional. Para um segmento linear, com $C(t) = C_b.(1-t) + C_f.t$ e $\tau(t) = \tau_b.(1-t) + \tau_f.t$, a solução da integral é:

$$\begin{aligned}
 I(D) &= \begin{cases} I_0 e^{-D \frac{\tau_b + \tau_f}{2}} + C_f - C_b e^{-D \frac{\tau_b + \tau_f}{2}} + \\ \frac{(C_b - C_f)}{\sqrt{D(\tau_b - \tau_f)}} e^{\frac{D}{2(\tau_b - \tau_f)} \tau_f^2} \sqrt{\frac{\pi}{2}} \cdot \\ \left[\operatorname{erf} \left(\tau_b \frac{\sqrt{D}}{\sqrt{2(\tau_b - \tau_f)}} \right) - \operatorname{erf} \left(\tau_f \frac{\sqrt{D}}{\sqrt{2(\tau_b - \tau_f)}} \right) \right] \end{cases} & (tb > tf) \\
 I(D) &= \begin{cases} I_0 e^{-D \frac{\tau_b + \tau_f}{2}} + C_f - C_b e^{-D \frac{\tau_b + \tau_f}{2}} + \\ \frac{(C_b - C_f)}{\sqrt{D(\tau_f - \tau_b)}} e^{\frac{-D}{2(\tau_f - \tau_b)} \tau_f^2} \sqrt{\frac{\pi}{2}} \cdot \\ \left[\operatorname{erfi} \left(\tau_f \frac{\sqrt{D}}{\sqrt{2(\tau_f - \tau_b)}} \right) - \operatorname{erfi} \left(\tau_b \frac{\sqrt{D}}{\sqrt{2(\tau_f - \tau_b)}} \right) \right] \end{cases} & (tb < tf) \\
 I(D) &= \begin{cases} I_0 e^{-\tau D} + C_b \left(\frac{1}{\tau D} - \frac{e^{-\tau D}}{\tau D} - e^{-\tau D} \right) + \\ C_f \left(1 + \frac{e^{-\tau D}}{\tau D} - \frac{1}{\tau D} \right) \end{cases} & (tb = tf)
 \end{aligned}
 \tag{2.12}$$

Uma função de transferência genérica poder ser discretizada em segmentos lineares, e a solução da eq. (2.6) aproximada pela composição das contribuições calculadas pela integração analítica desses segmentos. Essa abordagem apresenta uma margem de erro bem menor do que a discretização em segmentos com coeficiente de extinção e intensidade emitida constantes, como ilustrado na Figura 8. A utilização de segmentos lineares representa uma forma conveniente para a

especificação da função de transferência (Williams et al., 1998; Moreland et al., 2004).

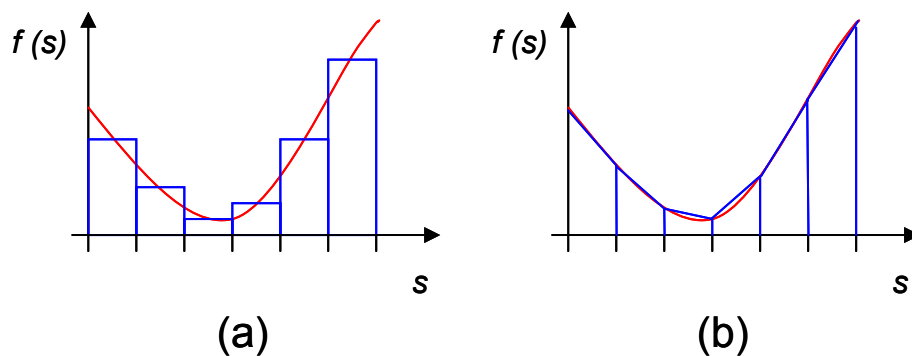


Figura 8 – Aproximação de uma função por segmentos constantes (a) e segmentos lineares (b).

2.3. Aplicação da função de transferência

Uma forma simples e direta de utilizar uma função de transferência arbitrária na visualização volumétrica de malhas de tetraedros consiste em resolver numericamente a eq. (2.6) no momento em que cada tetraedro é desenhado. Podemos convenientemente considerar que a intensidade de fundo I_0 é nula. Assim, a solução da eq. (2.6) resulta em um valor de cor associada e outro de opacidade, que correspondem à contribuição do tetraedro para a geração da imagem final, ao longo de um raio. Isto é equivalente à contribuição de um tetraedro (Figura 9) que possui cor e opacidade constantes, com valores iguais aos da solução obtida. Assim, supondo que os tetraedros são desenhados segundo uma ordenação (de trás para a frente, por exemplo), a eq. (2.9) pode ser utilizada para compor sucessivamente as contribuições de cada um, produzindo a imagem final.

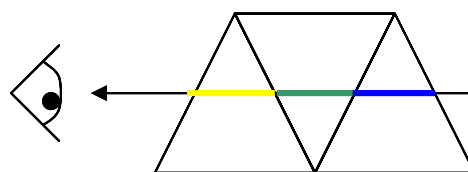


Figura 9 – Composição das contribuições de três tetraedros ao longo de um raio em direção ao observador.

Entretanto, o custo computacional de resolver numericamente a eq. (2.6) para cada posição desenhada é alto, inviabilizando a visualização interativa. Duas abordagens para contornar esse problema são apresentadas nas próximas seções: *pré-integração da função de transferência* e *integração de segmentos lineares*. Cada uma possui algumas vantagens e desvantagens.

2.3.1. Pré-integração

Essa técnica, apresentada por Roettger et al. (2000), consiste na integração da função de transferência em uma etapa de pré-processamento, antes da visualização da malha de tetraedros. A integração resulta em valores de cor associada e opacidade, que são armazenados em uma textura 3D (tabela tridimensional).

Considerando um raio que parte do observador e atravessa um tetraedro linear, como na Figura 10, e dada uma função de transferência cujo domínio é um campo escalar, a integral de renderização de volume (eq. 2.6) pode ser expressa em função de três variáveis:

- s_f : escalar da posição mais próxima do observador (entrada do raio no tetraedro);
- s_b : escalar da posição mais distante do observador (saída do raio do tetraedro);
- l : distância percorrida pelo raio no interior do tetraedro.

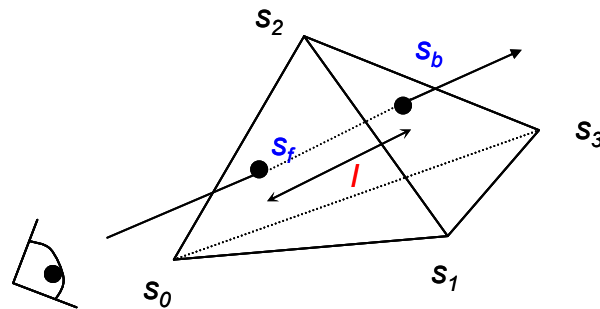


Figura 10 – Um raio que parte do observador e atravessa um tetraedro. Os escalares de entrada e saída, s_f e s_b , são interpolados a partir dos escalares associados aos vértices do tetraedro, enquanto que l é a distância entre a posição de entrada e saída do raio.

Em um tetraedro linear, o campo escalar varia linearmente em relação à distância l percorrida pelo raio, entre as posições de valores s_f e s_b . Assim, a eq. (2.2), relativa à opacidade do tetraedro, e o segundo termo da eq. (2.6), que resulta na cor associada C' , podem ser reescritos da seguinte forma (Engel et al., 2001), em função de (s_f, s_b, l) :

$$\alpha = 1 - e^{-\int_0^l \tau((1-t).s_f + t.s_b).l.dt} \quad (2.13)$$

$$C' = \int_0^l C((1-t).s_f + t.s_b). \tau((1-t).s_f + t.s_b). e^{-\int_0^t \tau((1-t').s_f + t'.s_b).l.dt'} l.dt \quad (2.14)$$

Dessa forma, a função de transferência deve ser integrada para cada posição (s_f, s_b, l) da textura 3D (Figura 11), de dimensões $m \times n \times d$, utilizando simplesmente somas de Riemann (Malta et al., 2002) ou qualquer outro método de integração numérica. Os valores intermediários são aproximados pela interpolação linear da cor associada e da opacidade armazenadas, feita automaticamente pela placa gráfica durante o mapeamento de textura. Para o mapeamento de textura, é conveniente que s_f, s_b e l sejam normalizados para o intervalo $[0,1]$.

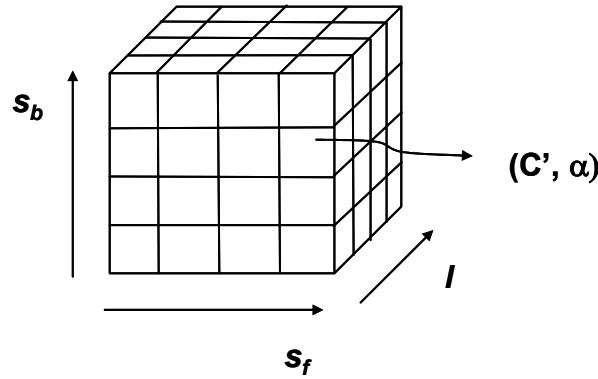


Figura 11 – Textura 3D contendo valores de cor associada e opacidade (C', α) , resultantes da pré-integração da função de transferência, em função dos escalares de entrada e saída e a distância do raio (s_f, s_b, l) .

A pré-integração apresenta dois problemas. O primeiro é relacionado à amostragem da função de transferência pré-integrada, principalmente para funções com altas frequências. Se a resolução da textura 3D for muito baixa, podem ser notados erros visuais, devido à baixa amostragem da função de transferência. Por outro lado, texturas com alta resolução ocupam muito espaço na

memória da placa gráfica, apesar de que este é um problema cada vez menor para as placas modernas.

O segundo problema está relacionado à atualização da função de transferência. O custo da integração numérica é $O(ns_f * ns_b * nl * npassos)$, sendo ns_f , ns_b e l , respectivamente, as dimensões da textura 3D quanto às coordenadas s_f , s_b e l , e $npassos$ o número de passos da integração da eq. (2.6) para uma dada posição (s_f, s_b, l) . Dependendo do tamanho da textura, a atualização pode levar vários segundos.

Uma grande limitação da pré-integração é que essa técnica é restrita a funções de transferência unidimensionais. Uma função de transferência multidimensional necessitaria de uma textura de dimensão superior a três, o que representa um custo de armazenamento muito alto, além desse tipo de textura não ser suportado pelas placas gráficas atuais.

Guthe et al. (2002) apresentam uma aproximação que permite reduzir a dimensão da textura para dois, sem que seja desprezada a variação do comprimento l para cada valor de (s_f, s_b) . Isto é feito, primeiramente, separando a eq. (2.13), o que permite armazenar $\tau'(s_f, s_b)$ em uma textura 2D e $\alpha(l, \tau')$ em uma textura 1D, de alta resolução:

$$\tau'(s_f, s_b) = \int_0^1 \tau(s_f + t(s_b - s_f)) dt \quad (2.15)$$

$$\alpha(l, \tau') = 1 - e^{-l \cdot \tau'} \quad (2.16)$$

Nas placas gráficas programáveis, o cálculo da opacidade α pode ser realizado diretamente no programa por fragmento. Para determinar a cor resultante, é utilizada uma aproximação polinomial da variação da cor ao longo da distância l , para cada valor de (s_f, s_b) . Ao invés de ser utilizada a cor associada resultante da eq. (2.14), o valor é “normalizado”, dividindo-se a cor associada pela opacidade na distância considerada:

$$C'(l) = \frac{C(l)}{\alpha(l)} \quad (2.17)$$

A cor normalizada será, então, aproximada por um polinômio de ordem n , sendo necessário resolver a eq. (2.14) para pelo menos $(n+1)$ valores de l , e a cor associada resultante será aproximada da seguinte forma:

$$C(s_f, s_b, l) \approx \alpha(s_f, s_b, l) \sum_{i=0}^n \frac{l^i}{l_{\max}^i} C'_i(s_f, s_b) \quad (2.18)$$

São necessárias $(n+1)$ texturas 2D em função de (s_f, s_b) para armazenar os coeficientes C'_i do polinômio. A vantagem dessa aproximação é que, utilizando texturas 2D, o custo de armazenamento é menor que no caso de texturas 3D, o que permite fazer uso de texturas de maior resolução. Utilizando texturas 3D, essa técnica poderia suportar alguns tipos de função de transferência bidimensionais, considerando um aumento no custo da pré-integração.

2.3.2.

Integração de segmentos lineares

A integração de segmentos lineares foi utilizada com sucesso no sistema HIAC, apresentado por Williams et al. (1998), para a visualização (não-interativa) de malhas de elementos finitos. Nesse sistema, a função de transferência é definida por segmentos lineares, como mostrado na Figura 12. Cada ponto na fronteira de um segmento linear é chamado *ponto de controle* e define uma iso-superfície plana, para o caso de um tetraedro linear. A técnica consiste em dividir cada tetraedro em “fatias” nas quais a variação da função de transferência é linear, ou seja, entre duas iso-superfícies. O resultado de cada fatia é determinado pela integração exata do segmento linear (eq. 2.12) e composto com as contribuições anteriores.

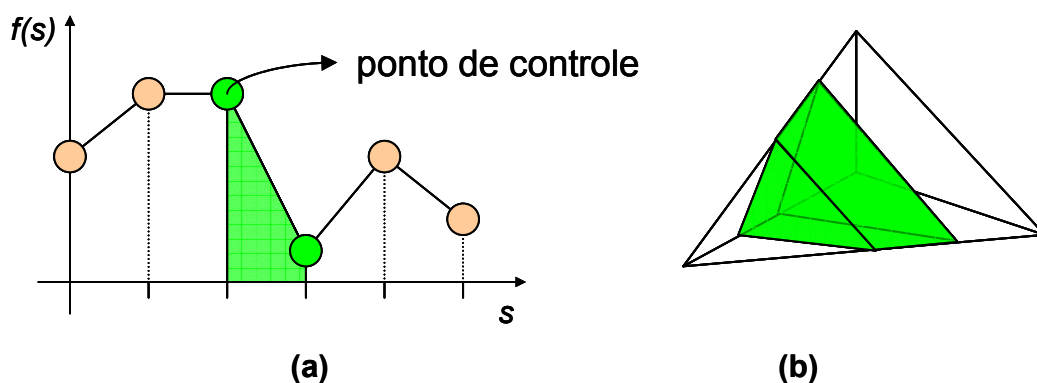


Figura 12 – (a) Função de transferência definida por segmentos lineares. (b) “Fatia” de um tetraedro limitada pelos dois pontos de controle de um segmento linear.

Recentemente, Moreland & Angel (2004) adaptaram essa abordagem para a visualização interativa de tetraedros lineares, com base em projeção de células

(seção 3.1). Para isso, exploram as placas gráficas programáveis para dividir os tetraedros em fatias durante a renderização, e a integração exata dos segmentos lineares é feita utilizando consultas a uma textura 2D contendo informações pré-calculadas.

Considerando um tetraedro linear por onde passam duas iso-superfícies definidas pela função de transferência e um raio que atravessa o tetraedro, os casos a serem considerados são os que aparecem na Figura 13.

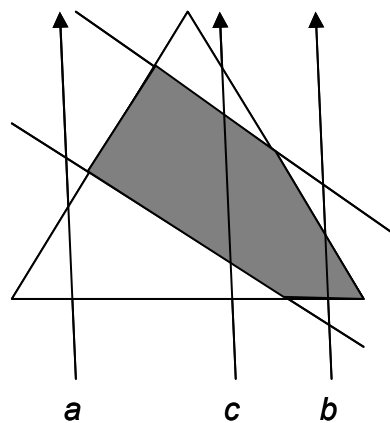


Figura 13 – Projeção de um tetraedro cortado por duas iso-superfícies definidas por pontos de controle da função de transferência. Os exemplos *a*, *b*, e *c* representam os casos fundamentais para um raio que atravessa o tetraedro.

O tetraedro deve ser desenhado uma vez para cada fatia definida por duas iso-superfícies (planos, no caso) que passam pelo tetraedro. Assim, os valores das iso-superfícies que limitam a fatia que está sendo desenhada devem ser computados na CPU e enviados para a placa gráfica. Para cada raio que atravessa o tetraedro, o programa por fragmento será responsável por determinar o caso em que o raio se encontra e se o respectivo fragmento deverá ou não ser desenhado. Supondo que a fatia a ser desenhada é a de cor *cinza*, no caso *a* o fragmento deverá ser descartado, enquanto que, no caso *b*, deverá ser desenhado e, no caso *c*, o raio deve ser projetado para os limites da fatia definidos pelos dois planos.

A contribuição mais importante de Moreland & Angel (2004) é uma nova formulação para a solução analítica da integral de renderização de volume (eq. 2.6), chamada de *pré-integração parcial*, para um segmento linear da função de transferência. A grande vantagem dessa formulação é que parte da solução da eq. (2.6) é tabulada, durante uma etapa de pré-processamento, em uma textura 2D, e é

independente da função de transferência. A solução completa pode, então, ser encontrada rapidamente através de uma consulta à textura e mais algumas poucas operações aritméticas no programa por fragmento.

Com base na eq. (2.6), é definido $C(t) = (1-t/D).C_b + (t/D).C_f$, e a equação fica:

$$I(D) = I_0.e^{-\int_0^D \tau(t)dt} + \int_0^D \left(C_b \left(1 - \frac{t}{D} \right) + C_f \cdot \frac{t}{D} \right) \tau(t).e^{-\int_t^D \tau(t')dt'} dt \quad (2.19)$$

A integração por partes resulta em:

$$I(D) = I_0.e^{-\int_0^D \tau(t)dt} + C_b \left(-e^{-\int_0^D \tau(t)dt} + \frac{1}{D} \int_0^D e^{-\int_t^D \tau(t')dt'} dt \right) + C_f \left(1 - \frac{1}{D} \int_0^D e^{-\int_t^D \tau(t')dt'} dt \right) \quad (2.20)$$

A partir da eq. (2.20), definem-se os seguintes termos:

$$\zeta(D, \tau(t)) \equiv e^{-\int_0^D \tau(t)dt} \quad (2.21)$$

$$\psi(D, \tau(t)) \equiv \frac{1}{D} \int_0^D e^{-\int_t^D \tau(t')dt'} dt \quad (2.22)$$

Substituindo as eq. (2.21) e (2.22) na eq. (2.20), chega-se à seguinte equação, que pode ser facilmente computada, dados ζ e ψ :

$$I(D) = I_0.\zeta(D, \tau(t)) + C_b(\psi(D, \tau(t)) - \zeta(D, \tau(t))) + C_f(1 - \psi(D, \tau(t))) \quad (2.23)$$

Assumindo-se a variação linear do coeficiente de extinção e definindo-se $\tau(t) = (1-t/D).\tau_b + (t/D).\tau_f$, chegamos a:

$$\zeta(D, \tau_f, \tau_b) = e^{-\int_0^D \left(\tau_b(1-\frac{t}{D}) + \tau_f \cdot \frac{t}{D} \right) dt} = e^{-\frac{D}{2}(\tau_b + \tau_f)} \quad (2.24)$$

$$\psi(D, \tau_f, \tau_b) = \frac{1}{D} \int_0^D e^{-\int_t^D \left(\tau_b(1-\frac{t'}{D}) + \tau_f \cdot \frac{t'}{D} \right) dt'} dt = \int_0^1 e^{-D \int_t^1 \left(\tau_b(1-t') + \tau_f t' \right) dt'} dt \quad (2.25)$$

A eq. (2.25) pode ainda ser reescrita em função de $(\tau_f D, \tau_b D)$, o que permite armazenar os valores computados em uma textura 2D:

$$\psi(\tau_f D, \tau_b D) = \int_0^1 e^{-\int_t^1 \left(\tau_b D(1-t') + \tau_f t' \right) dt'} dt \quad (2.26)$$

Para isso, ainda é conveniente normalizar os valores de ψ para o intervalo $[0,1)$, de forma a permitir a utilização de texturas convencionais. Isso pode ser feito através de uma mudança de variáveis, primeiramente definindo:

$$\gamma \equiv \frac{\tau.D}{\tau.D + 1} \quad (2.27)$$

que resulta em:

$$\tau.D = \frac{\gamma}{1 - \gamma} \quad (2.28)$$

e, finalmente substituindo a eq. (2.28) na eq. (2.26):

$$\psi(\gamma_f, \gamma_b) = \int_0^1 e^{-\int_t^1 \left(\frac{\gamma_b}{1-\gamma_b}(1-t') + \frac{\gamma_f}{1-\gamma_f}t' \right) dt'} dt \quad (2.29)$$

Dessa forma, a eq. (2.23) pode ser resolvida no programa por fragmento calculando-se diretamente a eq. (2.27) e consultando-se a textura com os valores pré-computados da eq. (2.29).

Uma vantagem da integração de segmentos lineares é que essa abordagem permite utilizar funções de transferência multidimensionais, já que a solução da eq. (2.23) independe da função de transferência, o que não pode ser feito com a pré-integração. Outra é a qualidade da imagem gerada, superior à obtida com a pré-integração.

Uma desvantagem é que o custo para se desenhar um tetraedro é proporcional ao número de iso-superfícies que passam por ele. Assim, dependendo da função de transferência e do tipo de dados, pode ser necessário dividir o tetraedro diversas vezes. Moreland & Angel (2004) reportam um aumento de 3% a 4% no número de tetraedros desenhados para cada ponto de controle da função de transferência, considerando algumas malhas com resultados reais.

2.3.3.

Iso-superfícies na placa gráfica

Roettger et al. (2000) apresentam uma forma de desenhar iso-superfícies de tetraedros lineares com o auxílio da placa gráfica, sem que seja necessário extraí-las explicitamente. Para isso, é realizada uma espécie de pré-integração simplificada da função de transferência, e o resultado é armazenado em uma

textura 2D. Consideremos um raio que atravessa um tetraedro, com s_f e s_b sendo, respectivamente, os valores de um campo escalar nas posições de entrada e saída do raio. Se $s_{iso1} = 0.4$, $s_{iso2} = 0.5$ e $s_{iso3} = 0.75$ são os valores de três iso-superfícies opacas, com as cores *VERMELHO*, *VERDE* e *AZUL* definidas por impulsos na função de transferência, então a integração do raio ao longo do tetraedro pode ser codificada como exemplificado na Figura 14. Cada posição (s_f, s_b) da textura contém a cor da primeira iso-superfície atravessada pelo raio. As áreas de cor preta representam as posições da textura para as quais nenhuma iso-superfície é atravessada pelo raio.

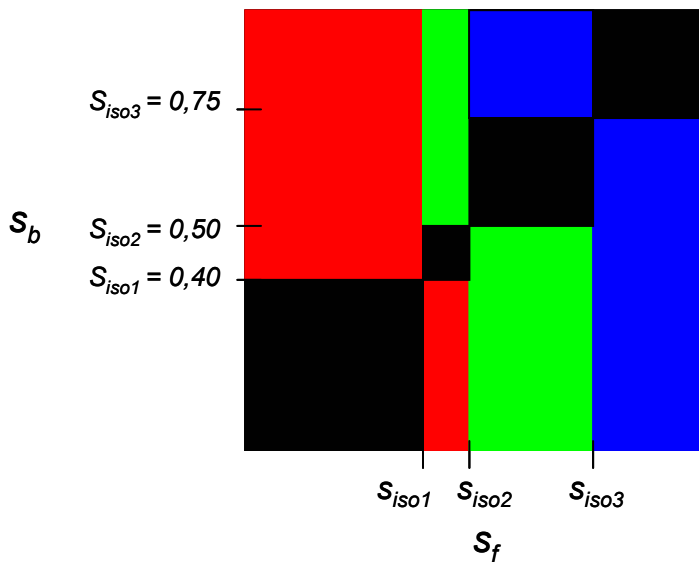


Figura 14 – Textura 2D contendo os valores de cor e opacidade da primeira iso-superfície atravessada por um raio, para cada par de coordenadas (s_f, s_b) .

Se, por exemplo, $s_f = 0,2$ e $s_b = 0,8$, a primeira iso-superfície atravessada pelo raio será $s_{iso1} = 0,4$. Assim, como as superfícies são opacas, a cor relativa a s_{iso1} , armazenada na textura 2D, será a cor final do fragmento desenhado. No caso de $s_f = 0,8$ e $s_b = 0,2$, a cor final será a de $s_{iso3} = 0,75$. Por outro lado, se $s_f = 0,6$ e $s_b = 0,7$, nenhuma iso-superfície será cortada pelo raio, e portanto nenhuma cor será associada àquela posição.

Esse método pode apresentar alguns erros visuais, tais como falhas e sobreposições de iso-superfícies junto à adjacência de dois tetraedros. Para contornar esse problema, Roettger et al. (2000) propõem incrementar a espessura das iso-superfícies, aumentando um pouco a sua área na textura 2D (Figura 15).

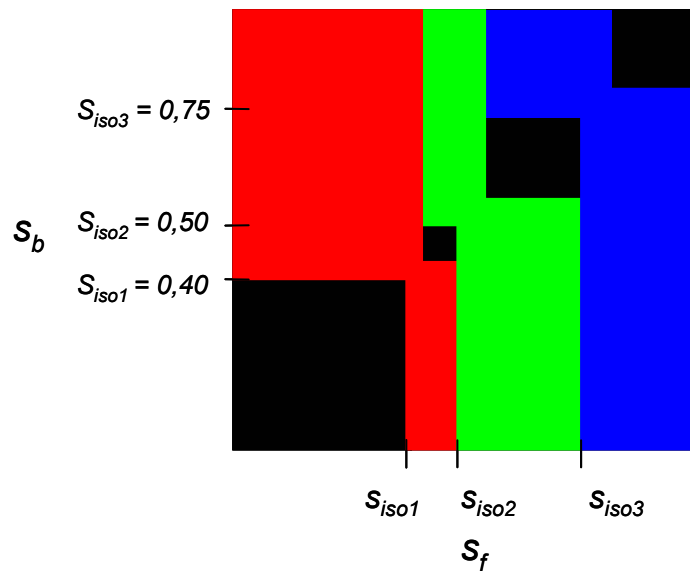


Figura 15 – As áreas relativas às iso-superfícies representadas na textura 2D são aumentadas para corrigir falhas visuais.

Uma vantagem dessa abordagem é que, independentemente do número de iso-superfícies que cortam um tetraedro, este pode ser desenhado em ordem constante ($O(1)$).